ZIMM AF

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application No.:	09/595,003	Examiner:	Pillai, Namitha
Filed:	June 13, 2000	Group/Art Unit:	2173
Inventor(s):		Atty. Dkt. No:	5150-44300
	NICOLAS VAZQUEZ, JEFFREY L. KODOSKY, RAM KUDUKOLI, KEVIN L. SCHULTZ, DINESH NAIR, and CHRISTOPHE CALTAGIRONE		
Title:	SYSTEM AND METHOD FOR AUTOMATICALLY GENERATING A GRAPHICAL PROGRAM TO IMPLEMENT A PROTOTYPE		

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to Commissioner for Patents, Alexandria, VA 22313-1450, on the date indicated below.

Jeffrey C. Hood

Signature — Date: 3/16/2005

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir/Madam:

Further to the Notice of Appeal filed August 26, 2004, Appellants present this Appeal Brief. Appellants respectfully request that this appeal be considered by the Board of Patent Appeals and Interferences.

03/22/2005 MAHMED1 00000051 501505 09595003
01 FC:1402 500.00 DA

I. REAL PARTY IN INTEREST

The subject application is owned by National Instruments Corporation, a corporation organized and existing under and by virtue of the laws of the State of Delaware, and having its principal place of business at 11500 N. MoPac Expressway, Bldg. B, Austin, Texas 78759-3504, as evidenced by the assignment recorded at Reel 014052, Frame 0095.

II. RELATED APPEALS AND INTERFERENCES

No other appeals, interferences or judicial proceedings are known which would be related to, directly affect or be directly affected by or have a bearing on the Board's decision in this Appeal.

III. STATUS OF CLAIMS

Claims 1-7, 9-37, 39-59, 61-74, and 76-90 are pending. Claims 1-7, 9-37, 39-59, 61-74, and 76-90 are rejected and are the subject of this Appeal Brief. A copy of claims 1-7, 9-37, 39-59, 61-74, and 76-90 as on appeal is included in the Claims Appendix attached hereto.

IV. STATUS OF AMENDMEMNTS

Appellant filed an Amendment after Notice of Appeal and prior to filing this Appeal Brief. That Amendment corrects a minor typographical error in claim 17. For the reason(s) explained in that Amendment, Appellant believes that that Amendment is entitled to entry under the standards set forth in MPEP §1207.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 is directed to a method of automatically creating a graphical program to perform an algorithm. A graphical program is a software program whose

source code is not textual in nature, but rather the source code is graphical in nature, e.g., a plurality of nodes or icons connected by wires.

One or more functions or operations of an algorithm or process may be recorded in response to user input, e.g., selecting or performing the one or more functions or operations of the algorithm or process. *See, e.g.,* Specification p. 24, lines 12-26; Figure 6 at 304. A graphical program which implements the one or more functions or operations of the algorithm or process may then be automatically generated in response to or based on the recorded functions. *See, e.g.,* Specification p. 24, lines 27-30; Figure 6 at 306. In automatically generating the graphical program, a plurality of interconnected nodes which visually indicate functionality of the graphical program may be automatically included in the graphical program without direct user input selecting the nodes. *See, e.g.,* Specification p. 24, line 30 - p. 25, line 3; p. 71, lines 12-14. In other words, the nodes are automatically included in the graphical program in an automatic or programmatic manner, without any direct user input required to select the nodes or place the nodes in the graphical program.

Independent claim 31 is directed to a system for automatically creating a graphical program to perform an algorithm. The system may include a processor or CPU. *See, e.g.,* Specification p. 20, line 10; Figure 3 at 160. The system may include a user input device. *See, e.g.,* Specification p. 76, line 3. The system may include a memory coupled to the processor which stores a prototyping environment application. *See, e.g.,* Specification p. 20, lines 13-16; Figure 3 at 166, 164, and 162. The prototyping environment application may be executable to perform the method of claim 1, the subject matter of which is summarized above.

Independent claim 53 is directed to a memory medium comprising program instructions executable to perform the method of claim 1, the subject matter of which is summarized above.

Independent claim 71 is directed to a method of creating a graphical program to perform an algorithm. The method includes creating a prototype in response to user input

where the prototype specifies the algorithm. *See, e.g.,* Specification p. 24, lines 5-14; Figure 6 at 302 and 304. The method also includes automatically generating the graphical program in response to the prototype. *See, e.g.,* Specification p. 24, lines 27-30; Figure 6 at 306. The graphical program implements the algorithm. *See, e.g.,* Specification p. 26, lines 1-3. The graphical program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program. *See, e.g.,* Specification p. 21, lines 19-21. In automatically generating the graphical program, the nodes are automatically included in the graphical program, i.e., are placed in the graphical program automatically. Hence, no direct user input is required to place the nodes in the graphical program, as they are automatically included. Also, no direct user input is required to select the nodes when they are being automatically included in the graphical program. *See, e.g.,* Specification p. 24, line 30 - p. 25, line 3; p. 71, lines 12-14.

Independent claim 81 is directed to a memory medium comprising program instructions executable to perform the method of claim 71, the subject matter of which is summarized above.

Independent claim 90 is directed to a memory medium comprising program instructions executable to record one or more functions in response to user input, where the one or more functions specify an algorithm. *See, e.g.,* Specification p. 24, lines 12-26; Figure 6 at 304. The memory medium further comprises program instructions executable to automatically generate a graphical program in response to the recorded one or more functions. *See, e.g.,* Specification p. 24, lines 27-30; Figure 6 at 306. The graphical program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program, and the graphical program implements the algorithm. *See, e.g.,* Specification p. 21, lines 19-21; Specification p. 26, lines 1-3. In automatically generating the graphical program, the program instructions are further executable to automatically generate graphical code in the graphical program without direct user input. *See, e.g.,* Specification p. 24, line 30 - p. 25, line 3; p. 71, lines 12-14.

In other words, the user is not required to manually specify the graphical code, but rather the graphical code is automatically generated by the program instructions.

Dependent claim 2 is directed to the functions being recorded in response to the functions being performed. In other words, the user may perform the one or more functions, e.g., on a computer system, and in response the performed one or more functions may be recorded. For example, the user may perform one or more image processing functions on an image, and the one or more image processing functions may be recorded. *See, e.g.,* Specification p. 24, lines 5-11.

Dependent claim 54 is directed to a memory medium which stores program instructions executable to perform functions which are recorded. Claim 54 also recites that the one or more functions are recorded in response to the functions being performed.

Dependent claim 21 is directed to a method where a user can specify code generation information. For example, in some embodiments, the user-specified code generation information may include a particular graphical programming language to use, a particular graphical programming development environment where the automatically generated graphical program may be used, a program type, a recorded script to use, an image source or type of image source, various parameters relating to an image hardware device, input specifying which image processing function parameters may be interactively specified or viewed, and/or desired parameter values associated with an image processing function, input and/or output parameters that are desired to be interactively changeable or viewable, among others. Furthermore, automatically generating the graphical program may utilize the user-specified code generation information. *See, e.g.,* Specification p. 30, line 24 - p. 34, line 11; Figures 15-20.

Dependent claim 45 is directed to a system which includes a prototyping environment application executable to perform the method of claim 21, and dependent claim 67 is directed to a memory medium which comprises program instructions executable to perform the method of claim 21. The subject matter of each of the claims 45 and 67 is summarized above.

Dependent claim 22 is directed to a method where a user can specify a type of graphical program to be created, and the graphical program is then automatically created in accordance with the specified type. *See, e.g.,* Specification p. 25, lines 4-17.

Dependent claim 46 is directed to a system which includes a prototyping environment application executable to perform the method of claim 22, and dependent claim 68 is directed to a memory medium which comprises program instructions executable to perform the method of claim 22. The subject matter of each of the claims 46 and 68 is summarized above.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1-7, 8-16, 21-37, 39-43, 45-59, 61-65, 67-74, and 76-90 are finally rejected under 35 U.S.C. 103(a) as being unpatentable over Morris et al. (U.S. Patent No. 5,862,372, hereinafter "Morris") and Lau-Kee et al. (U.S. Patent No. 5,631,974, hereinafter "Lau-Kee").

2. Claims 17-20, 44, and 66 are finally rejected under 35 U.S.C. 103(a) as being unpatentable over Morris and Lau-Kee in further view of Shi et al. (U.S. Patent No. 5,623,659, hereinafter "Shi").

VII. ARGUMENT

First Ground of Rejection:

Claims 1-7, 8-16, 21-37, 39-43, 45-59, 61-65, 67-74, and 76-90 are finally rejected under 35 U.S.C. 103(a) as being unpatentable over Morris and Lau-Kee. Appellant respectfully traverses this rejection for the following reasons. Different groups of claims are addressed under their respective subheadings.

Claims 1, 3, 5-7, 12, 13, 32, 42, 53, 55, 57-59, 64, 73, 77, 83, 86, and 90

Appellant respectfully submits that each of the independent claims 1, 53, and 90 recites one or more features not taught or suggested in Morris and Lau-Kee, taken both singly and in combination.

More specifically, the cited art does not teach or suggest: ". . .wherein said automatically generating the graphical program comprises automatically including the nodes in the graphical program, wherein said automatically including the nodes in the graphical program is performed without direct user input selecting the nodes (*emphasis added*)" as recited in claim 1. Thus, the cited art does not teach or suggest automatic creation of a graphical program as recited in the present claims.

More particularly, in Lau-Kee, the user is required to manually select and place an icon (node) in the program being created. Lau-Kee states "He then positions the icon. . ." (Lau-Kee col. 5, lines 57-60). Moreover, Lau-Kee's Abstract teaches and discloses that "A user can specify a sequence of image processing operations on the display (1) using the mouse (4), and interactively modify the sequence depending on the displayed image" (*emphasis added*). Lau-Kee also teaches and discloses "An advantage of the invention is that the user. . .may specify all the stages of the sequence by positioning them in the workspace prior to specifying any links between stages" (Lau-Kee col. 9, lines 20-26) (*emphasis added*).

Thus, Lau-Kee requires both direct user input to manually select the nodes and manual user interaction to include the node in the program being created. In other words, Lau-Kee teaches away from the present claims. Appellant respectfully notes "A reference may be said to teach away when a person of ordinary skill, upon reading the reference,. . .would be led in a direction divergent from the path that was taken by the applicant." *In re Haruna*, 249 F.3d 1327, 1335 (Fed. Cir. 2001) (quoting *Tec Air, Inc. v. Denso Mfg. Mich. Inc.,* 192 F.3d 1353, 1360 (Fed.Cir. 1999)) (*emphasis added*). Therefore, Lau-Kee is seen as teaching away from Appellant's present claims. Appellant submits that the current rejection does not make a *prima facie* case of obviousness. Further, Appellant respectfully notes "A *prima facie* case of obviousness can be rebutted if the applicant. . .can show 'that the art in any material respect taught away' from the claimed invention." *In re Haruna*, 249 F.3d 1327, 1335 (Fed. Cir. 2001) (quoting *In re*

Geisler, 116 F.3d 1465, 1469, 43 USPQ2d 1362, 1365 (Fed.Cir.1997) (quoting *In re Malagari,* 499 F.2d 1297, 1303, 182 USPQ 549, 553 (CCPA 1974))).

The Examiner relied on Morris as teaching various elements of the claims. However, Morris neither teaches nor suggests automatic creation of a graphical program as recited in the present claims. Rather, Morris teaches that a user of the system can manually develop an application. "Development of a complete application is accomplished by visually arranging, ordering, and interconnecting the objects without the necessity of writing any code" (Morris Abstract). More specifically, Morris teaches "the user of the system drags an object into the view. The system simultaneously generates the underlying script which reflects the object and the properties associated with the object" (Morris col. 6, lines 23-26) (*emphasis added*).

The Examiner stated in the Office Action mailed August 26, 2004: "Morris does not disclose that user input does not use the selection of the nodes. Lau-Kee discloses that the user may choose options from a menu wherein this menu allows for the user to choose a function, wherein upon this choice, the computer system would be responsible for finding the image icon represented by this function and placing this in the appropriate location, thereby not relying on user input to select the nodes (column 10, lines 32-43)" (*emphasis added*). However, this is not a correct characterization of Lau-Kee. The user's selection of a function from a menu in Lau-Kee is direct user input selecting a node for inclusion in the program. Therefore, Lau-Kee does not teach or suggest any type of automatic generation of a graphical program, and certainly does not teach or suggest automatic generation of a graphical program without direct user input selecting the nodes.

Further, regardless of the above, in Lau-Kee the user must still provide direct user input to move icons from the support area to the actual program. Appellant respectfully submits that ". . .placing this in the appropriate location. . ." as Lau-Kee teaches and discloses, in col. 10, lines 32-43, includes displaying "image processing icons" in a support area 11c. Appellant respectfully submits that support area 11c is not part of an image sequence process (program) in work space 10, as taught and disclosed in Lau-Kee upon reading Lau-Kee in its entirety.

Lau-Kee teaches that various icons or operations, e.g., icon 12a, may be selected by as user using a mouse from support areas 11a-11d to add functionality to the image processing sequence in work space 10 (Lau-Kee col. 4,line 61 - col. 10, line 56). In other words, displaying icons, which graphically represent image processing functions, in support area 11c does not affect the image processing sequence in work space 10. Thus, the display of program icons in support area 11c is accomplished by the user manually selecting the program icon from a menu, and in addition this does not create or add to the program at all. The user must further manually select one or more icons (e.g., icon 13a), which graphically represent image processing functions, from support area 11c and manually or directly add the selected icon(s) to the image processing sequence in work space 10 (Lau-Key col. 10, lines 9-56; Lau-Kee Figure 2).

The Examiner relied upon Lau-Kee to teach ". . .wherein said automatically generating the graphical program comprises automatically including the nodes in the graphical program, wherein said automatically including the nodes in the graphical program is performed without direct user input selecting the nodes (*emphasis added*)" as recited in claim 1. Appellant respectfully submits that neither Morris nor Lau-Kee teach or suggest this feature.

Appellant respectfully notes "To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 U.S.P.Q. 580 (C.C.P.A. 1974)" MPEP §2143.03 (*emphasis added*). Appellant respectfully submits that all the limitations of claim 1 are not taught or suggested by the prior art.

Thus, Appellant respectfully submits that a case of *prima facie* obviousness has not been established to reject claim 1. Accordingly, Appellant respectfully submits that, at least for the reasons presented, claim 1 is nonobvious and allowable.

Appellant respectfully notes "If an independent claim is nonobvious under 35 U.S.C. 103, then any claim depending therefrom is nonobvious. *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)" as stated in the MPEP §2143.03. Accordingly,

Appellant respectfully submits dependent claims 3, 5-7, 12, 13, and 32 are also nonobvious and allowable.

Claim 53 includes limitations similar to claim 1, specifically, the feature that ". . .wherein said automatically generating the graphical program comprises automatically generating graphical code in the graphical program without direct user input", and so the arguments presented above apply with equal force to claim 53, as well. More particularly, in each of the cited references, the user is required to manually select, include, arrange, and connect icons in the program being created. Accordingly, Appellant respectfully submits that, at least for the reasons presented, claim 53 is nonobvious and allowable.

Furthermore, Appellant respectfully submits dependent claims 55, 57-59, and 64 are also nonobvious and allowable.

Claim 90 includes limitations similar to claims 1 and 53, specifically, the feature that ". . .wherein, in automatically generating the graphical program, the program instructions are executable to automatically generate graphical code in the graphical program without direct user input", and so the arguments presented above apply with equal force to claim 90, as well. Accordingly, Appellant respectfully submits that claim 90 is nonobvious and allowable.

Appellant also respectfully submits that there is no teaching, suggestion, or motivation to combine Morris and Lau-Kee in either of the references or in the prior art. As held by the U.S. Court of Appeals for the Federal Circuit in *Ecolochem Inc. v. Southern California Edison Co.*, an obviousness claim that lacks evidence of a suggestion or motivation for one of skill in the art to combine prior art references to produce the claimed invention is defective as hindsight analysis. Thus, Appellant respectfully submits that it would be non-obvious to combine Morris and Lau-Kee.

Furthermore, the showing of a suggestion, teaching, or motivation to combine prior teachings "must be clear and particular. . .Broad conclusory statements regarding the teaching of multiple references, standing alone, are not 'evidence'." *In re Dembiczak*, 175 F.3d 994, 50 USPQ2d 1614 (Fed. Cir. 1999). The art must fairly teach or suggest to one to make the specific combination as claimed. That one achieves an improved result by making such a combination is no more than hindsight without an initial suggestion to make the combination. Appellant respectfully submits that there is no suggestion in the prior art for combining Morris and Lau-Kee. Further, even if Morris and Lau-Kee were properly combinable, which Appellant argues they are not, they would not produce the features of at least claims 1, 31, 53, 71, 81, and 90. Appellant respectfully submits claim 1, 31, 53, 71, 81, and 90 and claims respectively dependent thereon are allowable for at least the above reasons.

Claims 2 and 54

Appellant respectfully submits that neither Morris nor Lau-Kee teaches or suggests ". . .performing the one or more functions in response to user input. . ." and ". . .wherein said recording the one or more functions is performed in response to said performing the one or more functions" as recited by claim 2.

The Examiner stated in the Office Action mailed August 26, 2004: ". . .Morris discloses performing the function in response to user input, wherein the user dragging the objects is the function which is then recorded in response to the user's action for specifying the algorithm (column 3, lines 32-34)" (*emphasis added*).

Appellant respectfully submits that dragging an object is user input. However, the "functions" in this claim refer to the functions of the object itself in the program. For example, the user may perform one or more image processing functions on an image, and the one or more image processing functions may be recorded. In other words, ". . .the user dragging the objects. . ." is not ". . .performing the one or more functions in response to user input. . ." as recited by claim 2.

Thus, Appellant respectfully submits that claim 2 is patentably distinguished over both Morris and Lau-Kee, taken both singly and in combination. Accordingly, Appellant respectfully submits that, at least for one or more reasons presented, claim 2 is allowable.

Claim 54 includes limitations similar to claim 2, and so the arguments presented above apply with equal force to claim 54, as well. Appellant respectfully submits that, for at least the reasons presented above, claim 54 is patentably distinguished over both Morris and Lau-Kee, taken both singly and in combination, and is allowable.

Claims 9, 39, 61, 76, and 85

Appellant respectfully submits that claim 9 is allowable based, at least, on the arguments presented above for claims 1, 3, 5-7, 12, 13, 32, 42, 53, 55, 57-59, 64, 73, 77, 83, 86, and 90.

Furthermore, Appellant respectfully submits that neither Morris nor Lau-Kee teaches or suggests ". . .wherein said automatically generating the graphical program comprises automatically including and connecting the nodes in the graphical program without direct user input (*emphasis added*)."

Rather, Lau-Kee teaches and discloses that the user manually positions an icon (node) in the program "He then positions the icon 12b at a suitable point within the work space 10 using the user input device 4. . ." (Lau-Kee col. 5, lines 57-58), and manually or directly connecting icons (nodes) "By moving the cursor, the user can extend the link 14a since the display controller 2 updates the representation of the link 14a to keep one end at the cursor position. The user then moves the cursor to an input port 13b of the icon 12b representing the subtraction operation" (Lau-Kee col. 6, lines 1-5) (*emphasis added*).

In contrast, Appellant's invention as recited in claim 9 includes in pertinent part, ". . .wherein said automatically generating the graphical program comprises

automatically including and connecting the nodes in the graphical program without direct user input (*emphasis added*)." Thus, in this claim, the program icons or nodes are automatically included in the program and connected together by software, without any direct user input selecting the nodes or connecting them together. Neither Morris nor Lau-Kee teaches or suggests this feature. Thus, Appellant respectfully submits that claim 9 is patentably distinguished over both Morris and Lau-Kee, taken both singly and in combination. Accordingly, Appellant respectfully submits that, at least for the reasons presented, claim 9 is allowable.

Claims 39 and 61 include limitations similar to claim 9, and so the arguments presented above apply with equal force to these claims, as well. Appellant respectfully submits that, for at least the reasons presented above, claims 39 and 61 are patentably distinguished over both Morris and Lau-Kee, taken both singly and in combination, and are allowable.

Claims 76 and 85 includes limitations similar to claim 9, specifically each include the feature that ". . .wherein said automatically generating the graphical program comprises programmatically including and connecting the nodes in the graphical program without direct user input", and so the arguments presented above apply with equal force to claims 76 and 85, as well. Appellant respectfully submits that, for at least the reasons presented above, claims 76 and 85 are patentably distinguished over both Morris and Lau-Kee, taken both singly and in combination, and are allowable.

Claims 11, 40, and 63

Appellant respectfully submits that neither Morris nor Lau-Kee teaches or suggests the feature recited in claim 11:

> The method of claim 1,
>
> wherein the graphical program includes a block diagram portion and a user interface panel portion.

The Examiner relied upon "the palette in [Morris'] Figure 5" to teach "wherein the graphical program includes. . .a user interface panel portion" of Appellant's claim 11.

Appellant respectfully submits that the palette in Morris' Figure 5 is a part of an integrated development environment (IDE) which receives user input during manual creation of a script. More particularly, the IDE shown in Morris' Figure 5 is not part of a graphical program which comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program. Rather Morris teaches:

> FIG. 5 shows for a simple program all four views, Output, Map, Multitrack, and Workform simultaneously displayed in separate windows. In each view, the system of this invention permits development of the application using the same method of adding an object, interconnecting the object, and setting the objects properties. For instance, from a palette of objects, which displays a collection of icons that represent each object available to the system (such as the palette shown in FIG. 6), the user of the system drags an object into the view. The system simultaneously generates the underlying script which reflects the object and the properties associated with the object. (Morris col. 6, lines 15-26) (*emphasis added*)

Furthermore, Appellant respectfully submits that Morris neither teaches nor suggests that the script created, which is not a graphical program, includes a block diagram portion and a user interface panel portion.

In contrast, Appellant's invention as recited in claim 11 includes in pertinent part, ". . . wherein the graphical program includes a block diagram portion and a user interface panel portion (*emphasis added*)." Neither Morris nor Lau-Kee teaches or suggests this feature. Thus, Appellant respectfully submits that claim 11 is patentably distinguished over both Morris and Lau-Kee, taken both singly and in combination. Accordingly, Appellant respectfully submits that, at least for one or more reasons presented, claim 11 is allowable.

Claims 40 and 63 include limitations similar to claim 11, specifically each include ". . . wherein the graphical program includes a block diagram portion and a user interface portion (*emphasis added*)", and so the arguments presented above apply with equal force to these claims, as well. Appellant respectfully submits that, for at least the reasons presented above, claims 40, and 63 are patentably distinguished over both Morris and Lau-Kee, taken both singly and in combination, and are allowable.

Claims 14, 43, and 65

Appellant respectfully submits that neither Morris nor Lau-Kee teaches or suggests the combinations of features ". . .modifying the script to create a new script in response to user input after said creating the association. . ." and ". . .modifying the graphical program according to the new script to create a new graphical program" as recited in claim 14.

Rather, Morris' Abstract teaches and discloses: "The system generates as output a script listing the objects and their properties which is then executed by a separate run time program" (*emphasis added*). Furthermore, Morris also teaches and discloses that "For instance, from a palette of objects, which displays a collection of icons that represent each object available to the system (such as the palette shown in FIG. 6), the user of the system drags an object into the view. The system simultaneously generates the underlying script which reflects the object and the properties associated with the object" (Morris col. 6, lines 20-26) (*emphasis added*).

In other words, Morris teaches and discloses that the user drags objects into the view and then, in response and according to dragging the objects into the view, a script is simultaneously generated. In contrast, Appellant's invention as recited in claim 14 includes ". . .modifying the script to create a new script in response to user input after said creating the association. . ." and ". . .modifying the graphical program according to the new script to create a new graphical program". Thus, Morris teaches a direction divergent from the path that was taken by Appellant. Therefore, Morris is seen as teaching away from Appellant's present claims. Accordingly, Appellant respectfully

submits that a *prima facie* case of obviousness has not been established to reject claim 14. *See In re Haruna*, 249 F.3d 1327, 1335 (Fed. Cir. 2001) (outlining elements of a reference teaching away and rebutting a *prima facie* case of obviousness).

Accordingly, Appellant respectfully submits that, at least for one or more reasons presented, claim 14 and those dependent therefrom are allowable.

Claim 43 includes limitations similar to claim 14, specifically, the feature that ". . .wherein the graphical program creation program is executable to modify the graphical program according to the new script to create a new graphical program", and so the arguments presented above apply with equal force to claim 43, as well. Appellant respectfully submits that, for at least the reasons presented above, claim 43 is patentably distinguished over both Morris and Lau-Kee, taken both singly and in combination, and is allowable.

Claim 65 includes limitations similar to claim 14, and so the arguments presented above apply with equal force to claim 65, as well. Appellant respectfully submits that, for at least the reasons presented above, claim 65 is patentably distinguished over both Morris and Lau-Kee, taken both singly and in combination, and is allowable.

Claim 15

Appellant respectfully submits that neither Morris nor Lau-Kee teaches or suggests the combinations of features recited in claim 15:

> The method of claim 14,
>
> wherein said modifying the graphical program according to the new script uses the association between the script and the graphical program;
>
> wherein the association remains between the new script and the new graphical program.

Rather, Morris teaches and discloses "For instance, from a palette of objects, which displays a collection of icons that represent each object available to the system (such as the palette shown in FIG. 6), the user of the system drags an object into the view. The system simultaneously generates the underlying script which reflects the object and the properties associated with the object" (Morris col. 6, lines 20-26) (*emphasis added*).

In contrast, Appellant's invention as recited in claim 15 includes in pertinent part, ". . .wherein said modifying the graphical program according to the new script uses the association between the script and the graphical program. . ." (*emphasis added*). Neither Morris nor Lau-Key teach or suggest this feature. Thus, Appellant respectfully submits that claim 15 is patentably distinguished over both Morris and Lau-Kee, taken both singly and in combination. Accordingly, Appellant respectfully submits that, at least for one or more reasons presented, claim 15 is allowable.

Claim 16

Appellant respectfully submits that neither Morris nor Lau-Kee teaches or suggests the features of claim 16:

> The method of claim 14, further comprising:
>
> receiving user input indicating a desire to change the graphical program;
>
> displaying script information of the script;
>
> modifying the script information in response to user input; and
>
> modifying the graphical program after said modifying the script information.

Appellant respectfully submits that Morris teaches and discloses: "First, the system can be used to author applications using an entirely visual programming scheme (paradigm) which does not require the user to know or be able to write any specialized

code. Icons representing the objects (and accordingly their functionalities) may be placed (dragged) into an appropriate view" (Morris col. 3, lines 30-33) (*emphasis added*).

In other words, Morris teaches and discloses using "an entirely visual programming scheme" while Appellant invention recites ". . .modifying the script information in response to user input. . ." and ". . .modifying the graphical program after said modifying the script information" as recited in claim 16. ". Thus, Morris teaches a direction divergent from the path that was taken by Appellant. Therefore, Morris is seen as teaching away from Appellant's present claims. Accordingly, Appellant respectfully submits that a *prima facie* case of obviousness has not been established to reject claim 16. *See In re Haruna*, 249 F.3d 1327, 1335 (Fed. Cir. 2001) (outlining elements of a reference teaching away and rebutting a *prima facie* case of obviousness). In particular, Morris does not teach or suggest that user modification of a script can cause a corresponding graphical program to be automatically modified accordingly.

Appellant respectfully submits that, at least for one or more reasons presented, claim 16 is allowable.

Claims 21, 45, and 67

Appellant respectfully submits that neither Morris nor Lau-Kee teaches or suggests the combinations of features recited in claim 21:

> The method of claim 1, further comprising:
>
> receiving user input specifying code generation information;
>
> wherein said automatically generating the graphical program utilizes the code generation information.

Neither Morris nor Lau-Kee allows a user to specify code generation information this is then used to affect an automatic graphical program generation process. As noted above, Morris and Lau-Kee do not teach or suggest any type of automatic generation of a graphical program, and certainly do not allow a user to specify code generation

information to affect an automatic graphical program generation process. Thus, Appellant respectfully submits that claim 22 is patentably distinguished over both Morris and Lau-Kee, taken both singly and in combination.

Accordingly, Appellant respectfully submits that, at least for the reasons presented, claim 21 is allowable.

Claim 45 includes limitations similar to claim 21, specifically the features ". . .wherein the prototyping environment application is executable to receive user input specifying code generation information, wherein the code generation specifies information to use in generating the graphical program", and so the arguments presented above apply with equal force to claim 45, as well. Appellant respectfully submits that, for at least the reasons presented above, claim 45 is patentably distinguished over both Morris and Lau-Kee, taken both singly and in combination, and are allowable.

Claim 67 includes limitations similar to claim 21, and so the arguments presented above apply with equal force to claim 67, as well. Appellant respectfully submits that, for at least the reasons presented above, claim 67 is patentably distinguished over both Morris and Lau-Kee, taken both singly and in combination, and is allowable.

Claims 22, 23, 46, 47, and 68

Appellant respectfully submits that neither Morris nor Lau-Kee teaches or suggests the combination of features recited in claim 22:

> The method of claim 21,
>
> wherein the code generation information specifies a type of graphical program to create in response to the recorded one or more functions;
>
> wherein the graphical program is created in accordance with the specified graphical program type.

Nowhere does Morris or Lau-Kee teach or suggest that the user can specify code generation information that specifies a type of graphical program to create, wherein this type of graphical program is then automatically created. Thus, Appellant respectfully submits that claim 22 is patentably distinguished over both Morris and Lau-Kee, taken both singly and in combination.

The Examiner relied upon Morris col. 3, lines 5-15 to teach the features of claims 22, 46, and 68. Morris teaches and discloses: "Programs of the prior art each have their own object definition standards. That is, for an object to be incorporated into their system, it must meet that program's standards. No prior art program is available which will allow addition of objects not designed to its standard without the necessity of writing additional code to perform the interface to that object" (Morris col. 3, lines 12-18) (*emphasis added*). In other words, Morris teaches and discloses different object types but not different graphical program types as described in Appellant's Specification as summarized above on page 5 of this Appeal Brief. Further, as noted above, Morris does not teach any type of automatic generation of a graphical program, and certainly does not allow a user to specify a type of graphical program that is automatically created.

Accordingly, Appellant respectfully submits that, at least for one or more reasons presented, claim 22 and those dependent therefrom are allowable.

Claims 46 and 68 include limitations similar to claim 22, and so the arguments presented above apply with equal force to these claims, as well. Appellant respectfully submits that, for at least the reasons presented above, claims 46 and 68, and any claims respectively dependent therefrom, are patentably distinguished over both Morris and Lau-Kee, taken both singly and in combination, and are allowable.

Claims 24, 48, and 69

Appellant respectfully submits that neither Morris nor Lau-Kee teaches or suggests ". . .wherein said automatically generating the graphical program comprises enabling the graphical program to receive user input during program operation, wherein the user input specifies values for the specified one or more input

parameters. . .(*emphasis added*)" as recited in claim 24. Neither Morris nor Lau-Kee teaches or suggests automatic generation of a graphical program, and further do not teach or suggest that such an automatic generation of a graphical program would enable the automatically created graphical program to receive user input during program operation.

Rather, Morris describes an integrated development environment (IDE) which receives user input during creation of a script:

> . . .the user of the system drags an object into the view. The system [IDE] simultaneously generates the underlying script which reflects the object and the properties associated with the object. The user's double click on the icon representing the object presents a menu list of the properties for that object the settings of which can be changed according to the user's requirements. If a separate dialog box is available for a particular property, clicking on the property gets the user to the dialog box. As a property setting is defined, the system records that definition in the associated underlying script. In addition, the system user can elect to have all the four views synchronized so that each view is updated for every change made in any view to the underlying script. (Morris col. 6, lines 23-36) (*emphasis added*)

The above quoted language from Morris, which was relied upon by the Examiner, refers to dialog boxes used during program creation, not during program operation, i.e., execution of the program after it is created.

Thus, Appellant respectfully submits that claim 24 is patentably distinguished over both Morris and Lau-Kee, taken both singly and in combination. Accordingly, Appellant respectfully submits that, at least for one or more reasons presented, claim 24 is allowable.

Claims 48 and 69 include limitations similar to claim 24, and so the arguments presented above apply with equal force to these claims, as well. Appellant respectfully submits that, for at least the reasons presented above, claims 48 and 69 are patentably

distinguished over both Morris and Lau-Kee, taken both singly and in combination, and are allowable.

Claims 25, 49, 70, 80, and 89

Appellant respectfully submits that neither Morris nor Lau-Kee teaches or suggests that ". . . automatically generating the graphical program comprises generating portions of graphical code . . . and linking the portions of graphical code together" as recited in claim 25. Neither of the cited references teaches or suggests automatic code generation that comprises automatically linking portions of graphical code together.

Thus, Appellant respectfully submits that claim 25 is patentably distinguished over both Morris and Lau-Kee, taken both singly and in combination. Accordingly, Appellant respectfully submits that, at least for one or more reasons presented, claim 25 is allowable.

Claims 49, 70, 80, and 89 include limitations similar to claim 25, and so the arguments presented above apply with equal force to these claims, as well. Appellant respectfully submits that, for at least the reasons presented above, claims 49, 70, 80, and 89 are patentably distinguished over both Morris and Lau-Kee, taken both singly and in combination, and are allowable.

Claims 26, 27-29, 50 and 51

Appellant respectfully submits that claims 26 and 50 are allowable based, at least, on the arguments presented above for claims 25, 49, 70, 80, and 89.

Furthermore, Appellant respectfully submits that neither Morris nor Lau-Kee teaches or suggests ". . .wherein generating each portion of graphical code comprises connecting the node inputs and outputs together in order to implement the function with which the portion of graphical code is associated" as recited in claim 26.

Examiner relied upon Morris col. 5, lines 40-50 to teach this feature. Rather, Morris teaches and discloses "The four views are different ways to make and look at the same script. Each has its own special perspective and strength. FIG. 2 shows a Map view of a simple program in which the flow of the program and the flow of data is displayed" (Morris col. 5, lines 40-44) (*emphasis added*). Morris does not teach or disclose a portion of a graphical code; instead Morris teaches and discloses an entire "simple program". Appellant respectfully submits that Morris teaches and discloses a static view in col. 5, lines 40-50 and Morris' Figure 2 which does not teach connecting nodes to form a function.

In contrast, Appellant's invention as recited in claim 26 includes in pertinent part, ". . .wherein generating each portion of graphical code comprises connecting the node inputs and outputs together in order to implement the function with which the portion of graphical code is associated." Neither Morris nor Lau-Kee teaches or suggests this feature.

Thus, Appellant respectfully submits that claim 26 is patentably distinguished over both Morris and Lau-Kee. Accordingly, Appellant respectfully submits that, at least for the reasons presented, claim 26 is allowable.

Claim 50 includes limitations similar to claim 26 and so the arguments presented above apply with equal force to claim 50, as well. Appellant respectfully submits that, for at least the reasons presented above, claim 50 is patentably distinguished over both Morris and Lau-Kee, taken both singly and in combination, and is allowable.

Claims 30 and 52

Appellant respectfully submits that claims 30 and 52 are allowable based, at least, on the arguments presented above for claims 25, 49, 70, 80, and 89.

Furthermore, Appellant respectfully submits that neither Morris nor Lau-Kee teaches or suggests ". . .wherein generating the portion of graphical code that implements

a particular function utilizes the database information retrieved for the particular function (*emphasis added*)" as recited by claim 30.

Examiner relied upon "Mass Storage" of Morris' Figure 1 to teach this feature. Appellant respectfully notes that while a database may comprise various mass storage units, a mass storage unit does not teach or suggest using a database. Further, neither of the cited references teach or suggest that database information is used in automatically generating a portion of graphical code.

Thus, Appellant respectfully submits that claim 30 is patentably distinguished over both Morris and Lau-Kee, taken both singly and in combination. Accordingly, Appellant respectfully submits that, at least for one or more reasons presented, claim 30 is allowable.

Claim 52 includes limitations similar to claim 30 and so the arguments presented above apply with equal force to claim 52, as well. Appellant respectfully submits that, for at least the reasons presented above, claim 52 is patentably distinguished over both Morris and Lau-Kee, taken both singly and in combination, and is allowable.

Claims 31, 34, 41, 71, 74, 78, 79, 81, 84, 87, and 88

Appellant respectfully submits that claims 31, 34, 41, 71, 74, 78, 79, 81, 84, 87, and 88 are allowable based, at least, on the arguments presented above for claims 1, 3, 5-7, 12, 13, 32, 42, 53, 55, 57-59, 64, 73, 77, 83, 86, and 90.

Furthermore, Appellant respectfully submits that neither Morris nor Lau-Kee teaches or suggests the combination of features ". . .wherein the prototyping environment application is executable in response to the user input to store one or more functions in the memory, wherein the one or more functions specify the algorithm. . .(*emphasis added*)" and ". . .wherein, in automatically generating the graphical program, the prototyping environment application is executable to automatically include the nodes in

the graphical program without direct user input selecting the nodes (*emphasis added*)" as recited by claim 31.

Neither Morris nor Lau-Kee teaches or suggests automatic creation of a graphical program as discussed above with respect to claim 1.

Furthermore, Appellant respectfully notes that, upon reading Morris in its entirety, the terms "prototype", "prototyped", and "prototyping" appear nowhere in Morris.

In contrast, Appellant's invention as recited in claim 31 includes in pertinent part, ". . . the prototyping environment application is executable to automatically include the nodes in the graphical program without direct user input selecting the nodes. . .(*emphasis added*)" Neither Morris nor Lau-Kee teaches a prototyping environment application that automatically includes the nodes in a graphical program without direct user input selecting the nodes.

Accordingly, Appellant respectfully submits that, at least for the reasons presented, claim 31 is allowable.

Furthermore, Appellant respectfully notes "If an independent claim is nonobvious under 35 U.S.C. 103, then any claim depending therefrom is nonobvious. *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)" as stated in the MPEP §2143.03. Accordingly, Appellant respectfully submits that dependent claims 34 and 41 are allowable, as well.

Claim 71 includes limitations similar to claim 31, specifically, the features ". . .creating a prototype in response to user input, wherein the prototype specifies the algorithm. . .", ". . .automatically generating the graphical program in response to the prototype, wherein the graphical program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program, wherein the graphical program implements the algorithm. . .(*emphasis added*)", and ". . . wherein said

automatically generating the graphical program comprises automatically including the nodes in the graphical program, wherein said automatically including the nodes in the graphical program is performed without direct user input selecting the nodes (*emphasis added*)", and so the arguments presented above apply with equal force to claim 71, as well. Accordingly, Appellant respectfully submits that claim 71 is allowable.

Furthermore, Appellant respectfully submits that dependent claims 74, 78, and 79 are allowable, as well.

Claim 81 includes limitations similar to claim 31, specifically, the features ". . .creating a prototype in response to user input, wherein the prototype specifies the algorithm. . .", ". . .automatically generating the graphical program in response to the prototype, wherein the graphical program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program, wherein the graphical program implements the algorithm. . .(*emphasis added*)", and ". . .wherein said automatically generating the graphical program comprises automatically generating graphical code in the graphical program without direct user input (*emphasis added*)", and so the arguments presented above apply with equal force to claim 81, as well. Accordingly, Appellant respectfully submits that claim 81 is allowable.

Furthermore, Appellant respectfully submits that dependent claims 84, 87, and 88 are allowable, as well.

Claims 36 and 37

Appellant respectfully submits that claims 36 and 37 are allowable based, at least, on the arguments presented above for claims 31, 71, 78, 79, 81, 87, and 88.

Furthermore, Appellant respectfully submits that neither Morris nor Lau-Kee teaches or suggests ". . .wherein the graphical program creation program is executable to automatically generate the graphical program in response to said prototyping

environment application calling the graphical program creation program" as recited by claim 36.

The Examiner relied upon Morris col. 7, lines 4-7 to teach this feature. Appellant respectfully notes that Morris col. 7, lines 4-7 state: "Programs authored by the computer implemented application development system of this invention run faster under its run time program than to object oriented programs developed and run under Microsoft Visual Basic."

In contrast, Appellant's invention as recited in claim 36 includes in pertinent part, ". . .wherein the graphical program creation program is executable to automatically generate the graphical program in response to said prototyping environment application calling the graphical program creation program (*emphasis added*)." Neither Morris nor Lau-Kee teaches or suggests this feature, e.g., neither Morris nor Lau-Kee teach or suggest a prototyping environment application that calls a graphical program creation program to automatically create a graphical program. Thus, Appellant respectfully submits that claim 36 is patentably distinguished over both Morris andu-Kee, taken both singly and in combination. Accordingly, Appellant respectfully submits that, at least for the reasons presented, claim 36 is allowable.

Appellant respectfully submits that since claim 36 has been shown to be allowable, claim 37 is believed to be allowable, as well.

Second Ground of Rejection:

Claims 17-20, 44, and 66 are finally rejected under 35 U.S.C. 103(a) as being unpatentable over Morris and Lau-Kee in further view of Shi. Appellant respectfully traverses this rejection for the following reasons. Different groups of claims are addressed under their respective subheadings.

Claims 17-20, 44, and 66

Appellant respectfully submits that independent claims 1, 31, 53, 71, 81, and 90 have been argued above to overcome rejections under 35 U.S.C. 103 as being unpatentable over Morris and Lau-Kee. Appellant respectfully submits that, at least, based on the arguments presented above for claims 1, 3, 5-7, 12, 13, 32, 42, 53, 55, 57-59, 64, 73, 77, 83, 86, and 90, independent claims 1, 31, 53, 71, 81, and 90 are nonobvious in further view of Shi. Appellant respectfully notes "If an independent claim is nonobvious under 35 U.S.C. 103, then any claim depending therefrom is nonobvious. *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)" as stated in the MPEP §2143.03. Accordingly, Appellant respectfully submits that claims 17-20, 44, and 66 are nonobvious and allowable.

Furthermore, Appellant respectfully submits that Morris, Lau-Kee, and/or Shi nowhere teach or suggest ". . .locking the association between the script and the graphical program, wherein said locking prevents user editing of the graphical program" as recited by claim 17.

Thus, Appellant respectfully submits that claim 17 is patentably distinguished over Morris, Lau-Kee, and/or Shi.

The Examiner relied on Shi to teach locking an association between a script and a graphical program, where locking the association prevents user editing of the graphical program. However, Shi teaches user editing of a locked portion of a data set: ". . .the request is granted by allowing write access to the first portion [of the data set]. A write lock is placed on the first portion of the data set to prevent other users from changing it, while allowing other users read access to the first portion" (Shi Abstract) (*emphasis added*). In other words, Shi teaches that a first user has write or user editing access to the first portion of the data set while other users are prevented from writing to or editing the first portion of the data set.

Since Shi teaches user editing of a "locked" portion of a data set, Shi teaches can be seen as teaching away from locking an association which prevents user editing.

Accordingly, Appellant respectfully submits that a *prima facie* case of obviousness has not been established to reject claim 17. *See In re Haruna*, 249 F.3d 1327, 1335 (Fed. Cir. 2001) (outlining elements of a reference teaching away and rebutting a *prima facie* case of obviousness).

Furthermore, Shi nowhere teaches or suggests storing an association between a script and a graphical program in a portion of a data set. Shi is not related at all to graphical programming, and certainly does not teach or suggest locking an association between a script and a graphical program.

Further, as Shi does not relate to graphical programming, there is no teaching, suggestion or motivation to combine Shi with the other references as proposed by the Examiner. Appellant thus respectfully submits that there is no teaching, suggestion, or motivation to combine Morris, Lau-Kee, and Shi either in the references or in the prior art. As held by the U.S. Court of Appeals for the Federal Circuit in *Ecolochem Inc. v. Southern California Edison Co.*, an obviousness claim that lacks evidence of a suggestion or motivation for one of skill in the art to combine prior art references to produce the claimed invention is defective as hindsight analysis. Moreover, Appellant respectfully submits that it is nonobvious to combine Morris, Lau-Kee, and Shi.

Furthermore, the showing of a suggestion, teaching, or motivation to combine prior teachings "must be clear and particular. . .Broad conclusory statements regarding the teaching of multiple references, standing alone, are not 'evidence'." *In re Dembiczak*, 175 F.3d 994, 50 USPQ2d 1614 (Fed. Cir. 1999). The art must fairly teach or suggest to one to make the specific combination as claimed. That one achieves an improved result by making such a combination is no more than hindsight without an initial suggestion to make the combination. Appellant respectfully submits that there is no suggestion in the prior art for combining Morris, Lau-Kee, and Shi, and that even were the references combined, they would not produce the features of at least claims 17-20, 44, and 66. Appellant respectfully submits claim 17-20, 44, and 66 are allowable for at least the above reasons.

Thus, Appellant respectfully submits that a *prima facie* case of obviousness has not been established to reject claim 17. Accordingly, Appellant respectfully submits that, at least for the reasons presented, claim 17 and those dependent therefrom are allowable.

Claim 44 includes limitations similar to claim 17, specifically, the feature that ". . .wherein said locking prevents user editing of the graphical program", and so the arguments presented above apply with equal force to claim 44, as well. Thus, Appellant respectfully submits that claim 44 is patentably distinguished over Morris, Lau-Kee, and Shi, taken both singly and in combination. Accordingly, Appellant respectfully submits that, at least for the reasons presented, claim 44 is allowable.

Claim 66 includes limitations similar to claim 17, and so the arguments presented above apply with equal force to claim 66, as well. Thus, Appellant respectfully submits that claim 66 is patentably distinguished over Morris, Lau-Kee, and Shi, taken both singly and in combination. Accordingly, Appellant respectfully submits that, at least for the reasons presented, claim 66 is allowable.

VIII. CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1-7, 9-37, 39-59, 61-74, and 76-90 was erroneous, and reversal of Examiner's decision is respectfully requested.

The Commissioner is authorized to charge the appeal brief fee of $500.00 and any other fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5150-44300/JCH. This Appeal Brief is submitted with a return receipt postcard.

Respectfully submitted,

Jeffrey C. Hood
Reg. No. 35,198
Attorney for Appellant(s)

Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
(512) 853-8800

Date: March 16, 2005 JCH/IMF

IX. CLAIMS APPENDIX

The claims on appeal are as follows.

1. A method of creating a graphical program to perform an algorithm, the method comprising:

recording one or more functions in response to user input, wherein the one or more functions specify the algorithm; and

automatically generating the graphical program in response to the recorded one or more functions, wherein the graphical program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program, wherein the graphical program implements the algorithm;

wherein said automatically generating the graphical program comprises automatically including the nodes in the graphical program, wherein said automatically including the nodes in the graphical program is performed without direct user input selecting the nodes.

2. The method of claim 1, further comprising:

performing the one or more functions in response to user input;

wherein said recording the one or more functions is performed in response to said performing the one or more functions.

3. The method of claim 1,

wherein said recording the one or more functions comprises creating a prototype.

4. The method of claim 3,

wherein the prototype comprises a prototype in at least one of the disciplines from the group consisting of:

image processing, machine vision, image analysis, process control, industrial automation, test and measurement, simulation, workflow processes, and robotics.

5. The method of claim 1,

wherein said recording the one or more functions is performed in response to user input received via a graphical user interface (GUI).

6. The method of claim 5,

wherein the graphical user interface is associated with a prototyping environment application.

7. The method of claim 5,

wherein the user input comprises selecting the functions from one or more of a menu or palette.

8. (Cancelled)

9. The method of claim 1,

wherein said automatically generating the graphical program comprises automatically including and connecting the nodes in the graphical program without direct user input.

10. The method of claim 1, further comprising:

executing the graphical program to perform the algorithm.

11. The method of claim 1,

wherein the graphical program includes a block diagram portion and a user interface panel portion.

12. The method of claim 1,

wherein the graphical program is a graphical data flow program.

13. The method of claim 1,

wherein said automatically generating the graphical program comprises automatically including one or more nodes corresponding to respective ones of the one or more functions in the graphical program.

14. The method of claim 1, wherein the recorded one or more functions comprise a script, the method further comprising:

creating an association between the script and the graphical program;

modifying the script to create a new script in response to user input after said creating the association; and

modifying the graphical program according to the new script to create a new graphical program.

15. The method of claim 14,

wherein said modifying the graphical program according to the new script uses the association between the script and the graphical program;

wherein the association remains between the new script and the new graphical program.

16. The method of claim 14, further comprising:

receiving user input indicating a desire to change the graphical program;

displaying script information of the script;

modifying the script information in response to user input; and

modifying the graphical program after said modifying the script information.

17. The method of claim 14, further comprising:

creating an association between the script and the graphical program;

locking the association between the script and the graphical program, wherein said locking prevents user editing of the graphical program.

18. The method of claim 17, further comprising:

unlocking the association between the script and the graphical program in response to user input after said locking;

directly changing the graphical program in response to user input after said unlocking.

19. The method of claim 18,

wherein said unlocking removes the association between the script and the graphical program.

20. The method of claim 17, further comprising:

modifying the graphical program in response to user input after said generating the graphical program and after said creating the association between the script and the graphical program;

determining if an association exists between the script and the graphical program in response to said modifying the graphical program; and

removing the association between the script and the graphical program in response to said modifying.

21. The method of claim 1, further comprising:

receiving user input specifying code generation information;

wherein said automatically generating the graphical program utilizes the code generation information.

22. The method of claim 21,

wherein the code generation information specifies a type of graphical program to create in response to the recorded one or more functions;

wherein the graphical program is created in accordance with the specified graphical program type.

23. The method of claim 22,

wherein the graphical program type specifies a particular graphical programming environment;

wherein the graphical program is created in a file format that is usable by the particular graphical programming environment.

24. The method of claim 21,

wherein a plurality of parameters are associated with the one or more functions, wherein each parameter is an input parameter which provides input to a function or an output parameter which accepts output from a function;

wherein the code generation information specifies one or more of the input parameters which are desired to be interactively changeable or one or more of the output parameters which are desired to be interactively viewable;

wherein said automatically generating the graphical program comprises enabling the graphical program to receive user input during program operation, wherein the user input specifies values for the specified one or more input parameters;

wherein said automatically generating the graphical program comprises enabling the graphical program to display output during program operation, wherein the output indicates values for the specified one or more output parameters.

25. The method of claim 1,

wherein said automatically generating the graphical program comprises:

generating portions of graphical code, wherein each portion of graphical code implements one of the functions;

linking the portions of graphical code together.

26. The method of claim 25,

wherein each portion of graphical code includes one or more graphical program nodes, wherein each node has one or more inputs or outputs;

wherein generating each portion of graphical code comprises connecting the node inputs and outputs together in order to implement the function with which the portion of graphical code is associated.

27. The method of claim 26,

wherein linking a first portion of graphical code to a second portion of graphical code comprises connecting an output of a node in the first portion of graphical code to an input of a node in the second portion of graphical code.

28. The method of claim 26,

wherein at least one of the functions has an associated input parameter;

wherein each portion of code that implements a function that has an associated input parameter includes a node that has an input for receiving a value for the input parameter;

wherein each portion of code that implements a function that has an associated input parameter includes a leaf node that has an output for providing a value for the input parameter;

wherein the leaf node output for providing the parameter value is connected to the node input for receiving the parameter value.

29. The method of claim 26,

wherein at least one of the functions has an associated output parameter;

wherein each portion of code that implements a function that has an associated output parameter includes a node that has an output for providing a value for the output parameter;

wherein each portion of code that implements a function that has an associated output parameter includes a leaf node that has an input for receiving a value for the output parameter;

wherein the leaf node input for receiving the parameter value is connected to the node output for providing the parameter value.

30. The method of claim 25, further comprising:

for each function, retrieving information associated with the function from a database;

wherein generating the portion of graphical code that implements a particular function utilizes the database information retrieved for the particular function.

31. A system for creating a graphical program to perform an algorithm, the system comprising:

a processor;

a memory coupled to the processor which stores a prototyping environment application;

a user input device which receives user input;

wherein the prototyping environment application is executable in response to the user input to store one or more functions in the memory, wherein the one or more functions specify the algorithm;

wherein the prototyping environment application is executable to automatically generate a graphical program in response to the stored one or more functions, wherein the graphical program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program, wherein the graphical program implements the algorithm specified by the one or more functions;

wherein, in automatically generating the graphical program, the prototyping environment application is executable to automatically include the nodes in the graphical program without direct user input selecting the nodes.

32. The system of claim 31,

wherein said storing the one or more functions in the memory comprises creating a prototype.

33. The system of claim 32,

wherein the prototyping environment application is a prototyping environment application in at least one of the disciplines from the group consisting of:

image processing, machine vision, image analysis, process control, industrial automation, test and measurement, simulation, workflow processes, and robotics.

34. The system of claim 31,

wherein the prototyping environment application includes a graphical user interface (GUI);

wherein said storing the one or more functions in the memory is performed in response to user input received via the graphical user interface.

35. The system of claim 34,

wherein the user input comprises selecting the functions from one or more of a menu or palette.

36. The system of claim 31, further comprising:

a graphical program creation program stored in the memory;

wherein the prototyping environment application is executable to call the graphical program creation program;

wherein the graphical program creation program is executable to automatically generate the graphical program in response to said prototyping environment application calling the graphical program creation program.

37. The system of claim 36,

wherein the graphical program creation program is a graphical programming development environment application.

38. (Cancelled)

39. The system of claim 31,

wherein said automatically generating the graphical program comprises automatically including and connecting the nodes in the graphical program without direct user input.

40. The system of claim 31,

wherein the graphical program includes a block diagram portion and a user interface portion.

41. The system of claim 31,

wherein the graphical program is a graphical data flow program.

42. The system of claim 31,

wherein said automatically generating the graphical program comprises automatically including one or more nodes corresponding to respective ones of the one or more functions in the graphical program.

43. The system of claim 36,

wherein the stored one or more functions comprise a script;

wherein the memory stores an association between the script and the graphical program;

wherein the prototyping environment application is executable to utilize the association to modify the script to create a new script in response to user input;

wherein the graphical program creation program is executable to modify the graphical program according to the new script to create a new graphical program.

44. The system of claim 43,

wherein the memory stores an association between the script and the graphical program;

wherein the memory stores information specifying that the association between the script and the graphical program is locked, wherein said locking prevents user editing of the graphical program.

45. The system of claim 31,

wherein the prototyping environment application is executable to receive user input specifying code generation information, wherein the code generation specifies information to use in generating the graphical program.

46. The system of claim 45,

wherein the code generation information specifies a type of graphical program to create in response to the stored one or more functions;

wherein the graphical program is created in accordance with the specified graphical program type.

47. The method of claim 46,

wherein the graphical program type specifies a particular graphical programming environment;

wherein the graphical program is created in a file format that is usable by the particular graphical programming environment.

48. The system of claim 45,

wherein a plurality of parameters are associated with the functions, wherein each parameter is an input parameter which provides input to a function or an output parameter which accepts output from a function;

wherein the code generation information specifies one or more of the input parameters which are desired to be interactively changeable or one or more of the output parameters which are desired to be interactively viewable;

wherein said automatically generating the graphical program comprises enabling the graphical program to receive user input during program operation, wherein the user input specifies values for the specified one or more input parameters;

wherein said automatically generating the graphical program comprises enabling the graphical program to display output during program operation, wherein the output indicates values for the specified one or more output parameters.

49. The system of claim 31,

wherein said automatically generating the graphical program comprises:

generating portions of graphical code, wherein each portion of graphical code implements one of the functions;

linking the portions of graphical code together.

50. The system of claim 49,

wherein each portion of graphical code includes one or more graphical program nodes, wherein each node has one or more inputs or outputs;

wherein generating each portion of graphical code comprises connecting the node inputs and outputs together in order to implement the function with which the portion of graphical code is associated.

51. The system of claim 50,

wherein linking a first portion of graphical code to a second portion of graphical code comprises connecting an output of a node in the first portion of graphical code to an input of a node in the second portion of graphical code.

52. The system of claim 49,

wherein, for each function, information associated with the function is retrieved from a database;

wherein generating the portion of graphical code that implements a particular function utilizes the database information retrieved for the particular function.

53. A memory medium comprising program instructions executable to:

record one or more functions in response to user input, wherein the one or more functions specify an algorithm; and

automatically generate a graphical program in response to the recorded one or more functions, wherein the graphical program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program, wherein the graphical program implements the algorithm;

wherein said automatically generating the graphical program comprises automatically generating graphical code in the graphical program without direct user input.

54. The memory medium of claim 53, further comprising program instructions executable to:

perform the one or more functions in response to user input;

wherein said recording the one or more functions is performed in response to said performing the one or more functions.

55. The memory medium of claim 53,

wherein said recording the one or more functions comprises creating a prototype.

56. The memory medium of claim 55,

wherein the prototype comprises a prototype in at least one of the disciplines from the group consisting of:

image processing, machine vision, image analysis, process control, industrial automation, test and measurement, simulation, workflow processes, and robotics.

57. The memory medium of claim 53,

wherein said recording the one or more functions is performed in response to user input received via a graphical user interface (GUI).

58. The memory medium of claim 57,

wherein the graphical user interface is associated with a prototyping environment application.

59. The memory medium of claim 57,

wherein the user input comprises selecting the functions from one or more of a menu or palette.

60. (Cancelled)

61. The memory medium of claim 53,

wherein said automatically generating the graphical program comprises automatically including and connecting the nodes in the graphical program without direct user input.

62. The memory medium of claim 53, further comprising program instructions executable to:

execute the graphical program to perform the algorithm.

63. The memory medium of claim 53,

wherein the graphical program includes a block diagram portion and a user interface panel portion.

64. The memory medium of claim 53,

wherein said automatically generating the graphical program comprises automatically including one or more nodes corresponding to respective ones of the one or more functions in the graphical program.

65. The memory medium of claim 53, wherein the recorded one or more functions comprise a script, the memory medium further comprising program instructions executable to:

create an association between the script and the graphical program;

modify the script to create a new script in response to user input after said creating the association; and

modify the graphical program according to the new script to create a new graphical program.

66. The memory medium of claim 65, further comprising program instructions executable to:

create an association between the script and the graphical program;

lock the association between the script and the graphical program, wherein said locking prevents user editing of the graphical program.

67. The memory medium of claim 53, further comprising program instructions executable to:

receive user input specifying code generation information;

wherein said automatically generating the graphical program utilizes the code generation information.

68. The memory medium of claim 67,

wherein the code generation information specifies a type of graphical program to create in response to the recorded one or more functions;

wherein the graphical program is created in accordance with the specified graphical program type.

69. The memory medium of claim 67,

wherein a plurality of parameters are associated with the one or more functions, wherein each parameter is an input parameter which provides input to a function or an output parameter which accepts output from a function;

wherein the code generation information specifies one or more of the input parameters which are desired to be interactively changeable or one or more of the output parameters which are desired to be interactively viewable;

wherein said automatically generating the graphical program comprises enabling the graphical program to receive user input during program operation, wherein the user input specifies values for the specified one or more input parameters;

wherein said automatically generating the graphical program comprises enabling the graphical program to display output during program operation, wherein the output indicates values for the specified one or more output parameters.

70. The memory medium of claim 53,

wherein said automatically generating the graphical program comprises:

generating portions of graphical code, wherein each portion of graphical code implements one of the functions;

linking the portions of graphical code together.

71. A method of creating a graphical program to perform an algorithm, the method comprising:

creating a prototype in response to user input, wherein the prototype specifies the algorithm; and

automatically generating the graphical program in response to the prototype, wherein the graphical program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program, wherein the graphical program implements the algorithm;

wherein said automatically generating the graphical program comprises automatically including the nodes in the graphical program, wherein said automatically including the nodes in the graphical program is performed without direct user input selecting the nodes.

72. The method of claim 71,

wherein the prototype comprises a prototype in at least one of the disciplines from the group consisting of:

image processing, machine vision, image analysis, process control, industrial automation, test and measurement, simulation, telecommunications, workflow processes, and robotics.

73. The method of claim 71,

wherein the user input is received via a graphical user interface (GUI) associated with a prototyping environment application.

74. The method of claim 73,

wherein the user input comprises selecting one or more functions from one or more of a menu or palette.

75. (Cancelled)

76. The method of claim 71,

wherein said automatically generating the graphical program comprises programmatically including and connecting the nodes in the graphical program without direct user input.

77. The method of claim 71,

wherein said automatically generating the graphical program comprises automatically including one or more function nodes in the graphical program.

78. The method of claim 71,

wherein said creating the prototype in response to user input comprises creating a diagrammatic model of the algorithm.

79. The method of claim 71,

wherein said creating the prototype in response to user input comprises recording one or more functions in response to user input;

wherein the recorded one or more functions specify the algorithm.

80. The method of claim 79,

wherein said automatically generating the graphical program comprises:

generating portions of graphical code, wherein each portion of graphical code implements one of the functions;

linking the portions of graphical code together.

81. A memory medium comprising program instructions for creating a graphical program to perform an algorithm, wherein the program instructions are executable to implement:

creating a prototype in response to user input, wherein the prototype specifies the algorithm; and

automatically generating the graphical program in response to the prototype, wherein the graphical program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program, wherein the graphical program implements the algorithm;

wherein said automatically generating the graphical program comprises automatically generating graphical code in the graphical program without direct user input.

82. The memory medium of claim 81,

wherein the prototype comprises a prototype in at least one of the disciplines from the group consisting of:

image processing, machine vision, image analysis, process control, industrial automation, test and measurement, simulation, telecommunications, workflow processes, and robotics.

83. The memory medium of claim 81,

wherein the user input is received via a graphical user interface (GUI) associated with a prototyping environment application.

84. The memory medium of claim 83,

wherein the user input comprises selecting one or more functions from one or more of a menu and a palette.

85. The memory medium of claim 81,

wherein said automatically generating the graphical program comprises programmatically including and connecting the nodes in the graphical program without direct user input.

86. The memory medium of claim 81,

wherein said automatically generating the graphical program comprises automatically including one or more function nodes in the graphical program.

87. The memory medium of claim 81,

wherein said creating the prototype in response to user input comprises creating a diagrammatic model of the algorithm.

88. The memory medium of claim 81,

wherein said creating the prototype in response to user input comprises recording one or more functions in response to user input;

wherein the recorded one or more functions specify the algorithm.

89. The memory medium of claim 88,

wherein said automatically generating the graphical program comprises:

generating portions of graphical code, wherein each portion of graphical code implements one of the functions;

linking the portions of graphical code together.

90. A memory medium comprising program instructions executable to:

record one or more functions in response to user input, wherein the one or more functions specify an algorithm; and

automatically generate a graphical program in response to the recorded one or more functions, wherein the graphical program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program, wherein the graphical program implements the algorithm;

wherein, in automatically generating the graphical program, the program instructions are executable to automatically generate graphical code in the graphical program without direct user input.

X. EVIDENCE APPENDIX

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

XI. RELATED PROCEEDINGS APPENDIX

There are no related proceedings.